PERFORMANCE-BASED FACADE DESIGN TOOL: Approach for Automated and Multi-Objective Simulation and Optimization

Mahsa Minaei
University of Massachusetts Amherst

Abstract: Buildings have a considerable impact on the environment, and it is crucial to consider environmental and energy performance in building design. Buildings account for about 40% of the global energy consumption and contribute to over 30% of the CO_2 emissions. A large proportion of this energy is used for meeting occupants' thermal comfort in buildings, followed by lighting. The building facade forms a barrier between the exterior and interior environments; therefore, it has a crucial role in improving energy efficiency and building performance.

In this regard, decisionmakers are required to establish an optimal solution, considering multi-objective problems that are usually competitive and nonlinear, such as energy consumption, financial costs, environmental performance, occupant comfort, etc. Sustainable building design requires considerations of many design variables and multiple, often conflicting objectives, such as the initial construction cost, energy cost, energy consumption, and occupant satisfaction. One approach to address these issues is the use of building performance simulations and optimization methods.

This research presents a novel method for improving building facade performance, taking into consideration occupant comfort, energy consumption and energy costs. The research discusses development of a framework, which is based on multi-objective optimization and uses a Genetic Algorithm (GA) and machine learning in combination with building performance simulations. The framework utilizes the EnergyPlus simulation engine and custom scripts using Python programming to implement optimization algorithm analysis and decision support. The framework is automated in all steps: generating design scenarios, sending scenarios to the simulator, collecting the specific output, and decision making in optimization phase. So, the framework enhances the process of performance-based facade design, couples simulation and optimization packages, and provides a flexible and fast supplement in the facade design process by rapid generation of design alternatives.

The study describes the components and functionality of this framework in detail, as well as a two-step optimization technique, which is a new technique that combines GA and Machine Learning. This technique improves the framework speed, performance, and stability of an artificial neural network (ANN) and reduces the sensitivity.

The case study for a test cell presents, illustrating how the framework is used to test a variety of design possibilities and validation of this framework, as well as its application for facade design in different climates.

Keywords: Performance-based facade design, simulation-based optimization, machine learning, minimum viable product

INTRODUCTION

The buildings and building construction sectors combined are responsible for 36% of global final energy consumption and nearly 40% of total direct and indirect CO_2 emissions (IEA 2019). Energy demand from buildings and building construction continues to rise, driven by improved access to energy in developing countries, greater ownership and use of energy-consuming devices, and rapid growth in global buildings' floor area, at nearly 3% per year (IEA 2019). A large proportion of this energy is used for meeting occupants' thermal comfort in buildings, followed by lighting. The building facade forms a barrier between the exterior and interior environments, and has a crucial role in improving energy efficiency and building performance. Therefore, this research focuses on performance-based facade design, appropriate simulation and optimization tools, and methods for design analysis and support.

Building performance simulation (BPS) provides relevant design information by indicating potential (quantifiable) directions for design solutions. BPS tools and applications facilitate the process of design decision-making by providing quantifiable data about building performance. BPS tools are an integral part of the design process for energy efficient and high-performance buildings, since they help in investigating design options and assess the environmental and energy impacts of design decisions (Attia 2013). The important aspect is that simulation does not generate design solutions, instead, it supports designers by providing feedback on performance results of design scenarios.

Optimization is a method for finding a best scenario with highest achievable performance under certain constraints and variables. There are different methods for optimization, requiring use of computational simulation to achieve optimal solution, or sometimes requiring analysis or experimental methods to optimize building performance without performing mathematical optimization. In the BPS context, however, the term optimization generally indicates an automated process that is entirely based on numerical simulation and mathematical optimization (Nguyen 2014). Integrating BPS and optimization methods can form a process for selecting optimal solutions from a set of available alternatives for a given design problem, according to a set of performance criteria.

This paper focuses on developing a new framework for performance-based facade design. The framework considers energy consumption, occupant comfort, and energy cost optimality, and implements BPS and relevant optimization methods for performance-based facade design. The components and development of the framework are discussed in detail.

METHODOLOGY

The new framework for performance-based design approach, aiming to minimize building energy consumption and energy cost, while considering occupants' comfort level, was developed as part of this research. This is a modular framework, consisting of independent scripts that represent modules, steps, and functions of application under test. The modules are used in a hierarchical fashion to apply the framework, consisting of five steps:

1. Defining goals, performance criteria, facade design variables, and their properties, acceptable ranges for high-performance facade design.
2. Generating the database that includes all possible design scenarios based on the variables with permutation in Python and selected outputs after simulation in EnergyPlus. This is module 1.
3. Coupling Python script with simulation engine (EnergyPlus) to automatically perform simulations for scenarios from database (measurements methods) to quantify variables and generate the needed outputs. This is module 2.
4. Optimization phase by implementing Python script, genetic algorithm (GA) and Batch Normalization to evaluate outputs and find the optimal scenarios. This is module 3.

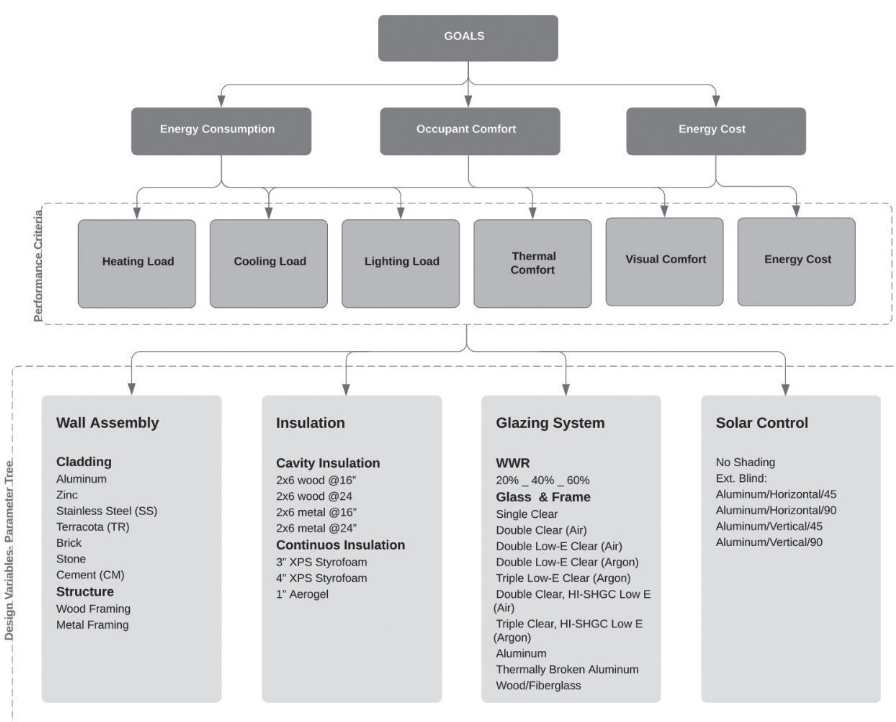

Figure 1: Conceptual diagram, showing components of the framework. (Author 2019)

5. Developing a front-end for user to test the framework and collect the data for next step, which is implementing deep learning after collecting enough data.

The next sections discuss the model development, then components of the framework and its implementation in detail will be illustrated.

Step 1: Defining Optimization Objectives, Performance Criteria and Facade Variables

Figure 1 shows the components of the framework. Performance-based facade design requires a holistic approach, considering performance indicators, such as energy performance and human comfort. These performance requirements (variables) must be quantified. The goals (optimization objectives) for this framework are to aid the design decision making process, where energy consumption and cost are minimized, and occupant comfort (thermal and visual) is maximized. The energy requirements for heating, cooling, and lighting of buildings are strongly driven by the performance of the facade, especially the glazing. The objectives for reducing energy consumption are to reduce heating, cooling, and lighting loads. Performance requirements (variables) to meet this objective are window to wall ratio (WWR), wall assembly, insulation, solar control, and glazing system. Performance-based facade design objectives that are related to human factors and contribute to occupant comfort and satisfaction in buildings include thermal comfort and visual comfort. The variables that relate to facade design include air temperature, mean radiant temperature, air movement, relative humidity, clothing layers and activity levels. The Predictive Mean Vote (PMV) suggested by Fanger (1970) predicts the effects of these six factors on thermal comfort. Predicted Percentage of Dissatisfied (PPD) persons predicts the percentage of people who would feel discomfort with certain thermal conditions. This research investigates how objectives are treated, what approach is more desirable and how to deal with constrained problems.

Step 2: Creating the Database (MYSQL)

After setting variables and parameters for facade design, all possible scenarios are generated using Python programming. With the permutation in Python script, design scenarios are generated and added to the database with a specific scenario ID. In this study, we have 38,400 scenarios to investigate for the test cell, described in the next section. After running simulation in EnergyPlus, all outputs in Step 3 are populated in this database with identical scenario ID. EnergyPlus provides a wide range of outputs, but, for this purpose, the following results are obtained:

- Cooling, heating, and lighting loads, Energy Use Intensity (EUI) for electricity and gas, PMV and PPD, and total energy costs for electricity and gas.
- Module 1 is responsible for generating all scenarios with defined variable and populating these scenarios in the database.
- Module 2 is responsible for automatically sending these scenarios to the simulation engine and for populating the selected outputs in the database.
- Data Flow Diagram (DFD) in figure 2 shows the overview of the framework system that represents the flow of data through this process.

The database manages all scenarios' inputs and outputs and is MYSQL, which is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL) for adding, accessing and managing content in the database. The advantages of this type of database for the purpose of this research is the quick processing time, proven reliability, open source, ease and flexibility of use.

Step 3: Coupling Python scripts with Simulation Engine (EnergyPlus)

EnergyPlus 8.5 is used in this research as an energy modeling engine. EnergyPlus has been chosen as a BPS tool for two main reasons: (a) this program allows reliable modeling of both building and HVAC systems, and (b) it works with text-based inputs and

System Range	Heating temperatures set point: upper & lower band	Cooling temperatures set point: upper & lower band 73.5-80.5 °F (23-27 °C)	Relative humidity set point: upper band 60 %RH lower band 30 %RH
Occupant Comfort Range	PMV range: -0.5 to +0.5	10% PPD (Predicted Percentage of Dissatisfied)	
Fixed Parameters	Type: Medium office Floor area: 1600 ft^2 Floor U-value: 0.10 Btu/ h ft^2 F Zones program: Open office	Operating hours for UDI analysis: 9 AM-5 PM Num. of people per area	Equipment load per area Infiltration rate per area

Table 1: List of fixed parameters

outputs, and these facilitate the interaction with Python scripts. EnergyPlus can investigate discussed variables as inputs and simulate envelope related outputs in the study. Thermal comfort is calculated based on PMV and PPD. The formulas for both PMV and PPD are built into EnergyPlus and their values can be obtained directly from the simulation output file.

Initial simulation test cell considered a single office space (40'x40'x10'), located in Atlanta, Georgia. The south-facing facade was used to develop different design scenarios, varying WWR, materials, glazing system, and shading control. Defining related parameters as inputs and setting data needed for outputs are the primary method for connecting design scenarios in the database with the simulation engine. Python script works as an interface to call scenarios from the database and to send them to the simulator. Each parameter must identify a well-defined relation with discussed variables, which reveals facade behavior in relation to performance aspects being analyzed.

Step 4: Optimization Phase by Implementing GA and Machine learning

In building optimization studies using GA, the computational time is generally reduced by two methods. First, is the use of a very simplified model instead of complete simulation. This simplification can cause inaccurate modeling of the building. The other method is selecting a very small size for GA populations or a relatively small number of generations. One efficient solution to reduce the computational time associated with GA algorithm is use of machine learning techniques to reduce time and increase the accuracy of the results. The machine learning used in this research is a combination of batch normalization, which is an Artificial Neural Network (ANN) technique, and flood fill algorithm. ANNs are effective methods that imitate the complex relationship of the network to solve multi objectives and non-linear problems. ANNs resemble the biological neural system, composed of layers of parallel neurons and weighted links. They learn the relationship between the input and output variables by studying previously recorded data. The layer that produces the network output are usually called the output layer, and all other layers are called hidden layers.

The optimization method in this study is a combination of GA and ANN, which is a machine learning technique. The GA in combination with flood fill algorithm and batch normalization creates a new technique to find a relation between the outputs, to assign weights, and dynamically adjust the target position to find optimal scenarios.

A batch normalization technique is used for the first phase of optimization. This is a technique for improving speed, performance, and stability of artificial neural network (ANN) by adjusting and scaling the activations. The batch normalization was introduced in 2015 by Ioffe and Szegedy. The intention behind batch normalization is to optimize network training. Several benefits of this methods are: faster training, higher learning rates, reduced sensitivity, and easier methods to initialize and produce better results (Hinton 2012). This technique, combined with a flood field algorithm, facilitates the optimization by sorting the highest indicators and decides which scenarios must be simulated.

The flood fill algorithm takes three parameters: start node, target, and replacement, and determines the

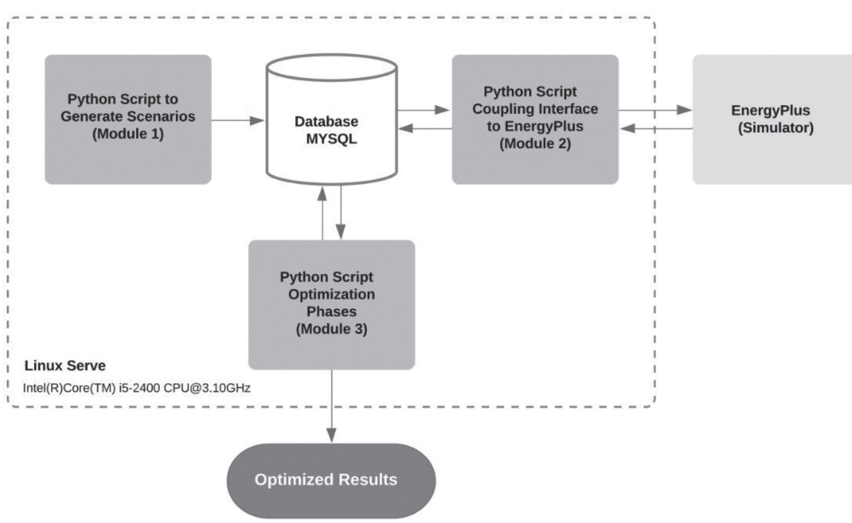

Figure 2: Data Flow Diagram (DFD) of the framework. (Author 2019)

area connected to our target. This algorithm facilitates the optimization by sorting the highest indicators and decides which scenarios must be simulated based on the specific scenario ID. Using this algorithm decreases the process time, because it is not necessary to simulate all scenarios—rather, only scenarios that are closer to the target. The comparison is based on the assigned indicator value. In a dynamic system, it is necessary to scale indicators to represent the impact of the indicators so as to configure following tasks and converge the results to the goal based on these scores. Figure 3 shows a sample for scoring total EUI electricity, EUI gas, PMV and energy cost indicators. These indicators work as fitness functions in genetic algorithms, which is a particular type of objective function to summarize and guide the simulations towards optimal design solutions. These indicators or fitness functions must correlate closely to the goals and must be computed quickly, because it needs to be iterated many times in order to produce usable results.

The initial population is generated randomly, based on the range of possible design scenarios. It is sent to the simulator to run the initial calculations, and then results are returned to the database to compare with the goals and standards. Then, design scenarios that have results closer to the goals are kept, and others are removed. In this framework, the goal is summation of three indicators, for energy consumption, comfort, and cost. The indicators are dynamically updated based on the range of results. The top left chart in figure 3 shows an example, where indicators from 6 to -3 are used for the initial test cell energy consumption results. Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. This method accelerates the simulation process and the results give us clusters of optimized scenarios for analysis in next phase of optimization. Figures 4 and 5 show how the optimization algorithm selects and sorts the fitted results for this framework.

Figure 4 shows the results before applying optimization for processing 2,061 scenarios and figure 5 shows the result of 18,103 scenarios with assigning the first step of optimization. In this case, we have 1,627 scenarios that scored 20 and more than 20 (1,591 scenarios at 20 and 36 more than 20). Using this process decreases the processing time, because it is not necessary to simulate all scenarios—rather, only scenarios that are closer to the target. After running all scenarios (38400 ID) and applying a batch normalization technique, 3,164 scenarios are selected. The next step focuses on comparison of results.

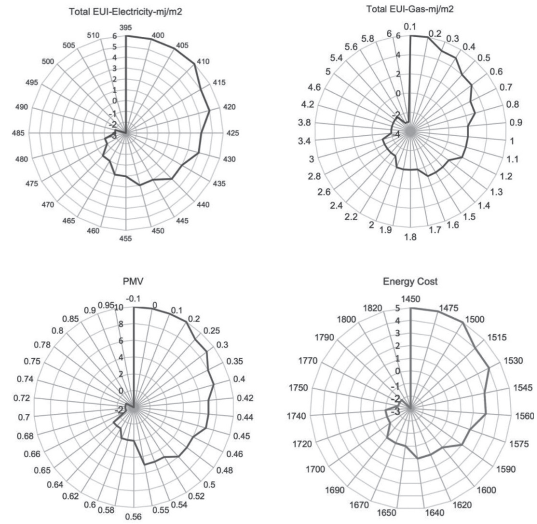

Figure 3: Total EUI-Electricity (MJ/m^2), EUI-Gas, PMV and Energy Cost indicator scores. (Author 2019)

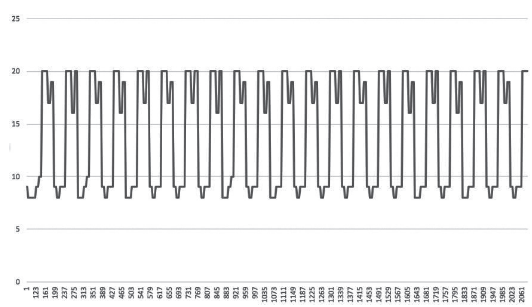

Figure 4: Total Indicators vs. Scenario IDs (for 2,061 scenarios). (Author 2019)

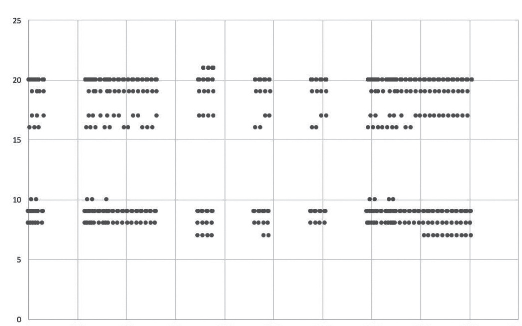

Figure 5: Total Indicators vs. Scenario IDs (for 18,103 scenarios). (Author 2019)

The next step of optimization is applying integrated correlation matrix clustering as a dropout technique, then comparing the results. This dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. Dropout refers to dropping out units

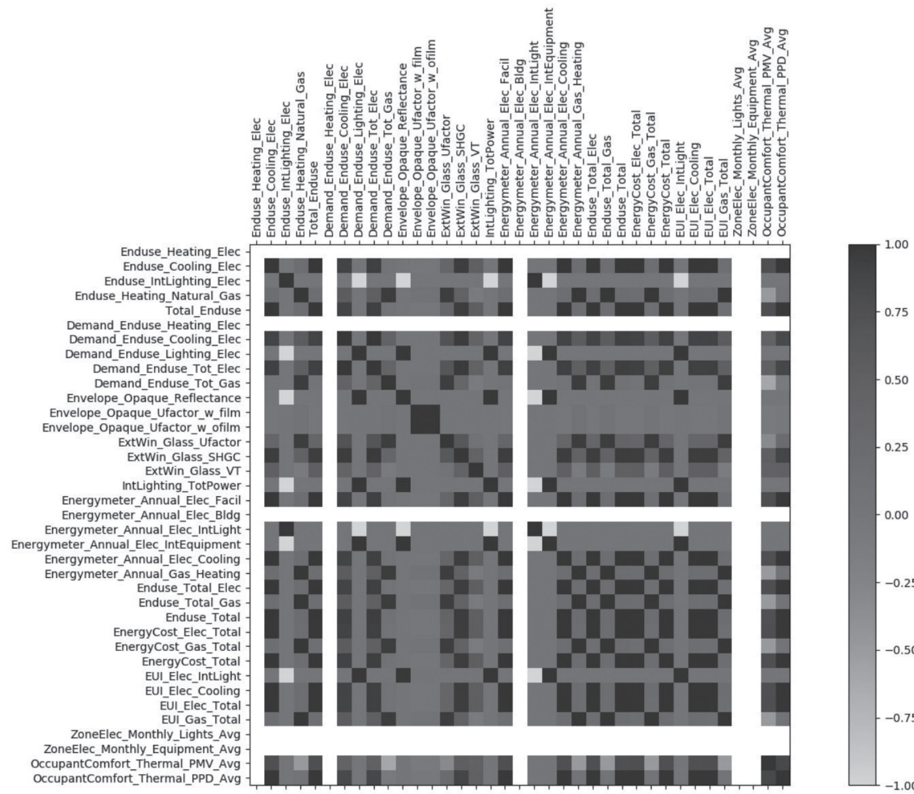

Figure 6: Correlation Matrix. (Author 2019)

(hidden or visible) in the neural network. In machine learning, correlation clustering provides a method for clustering a set of objects into an optimum number of clusters based on their similarity. So, in a correlation matrix, the relationship between the objects (variables) are known instead of the actual representations of the objects. Figure 6 shows the correlation matrix based on output data, integrated with optimization method to sort the results. Figures 7 and 8 show the final results of all scenarios with both techniques implemented. Figure 7 represents the first phase of optimization, and figure 8 shows the second phase after applying a correlation matrix and batch normalization. Results show that process time, performance, and accuracy are improved by using this method.

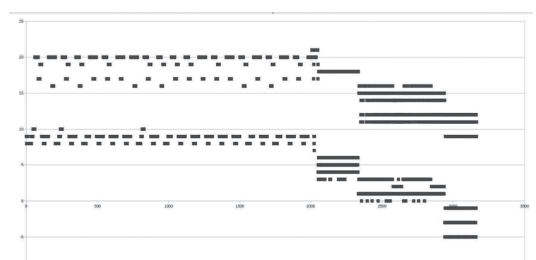

Figure 7: Results for all scenarios with applying batch normalization and flood field algorithm (Phase 1 optimization). (Author 2019)

Figure 8: Results for all scenarios with applying correlation matrix and cluster eliminating (Phase 2 optimization). (Author 2019)

Step 5: Developing a front-end for user to test the framework and collect the data for next step which is implementing deep learning after collecting enough data.

This research studies simulation-based optimization methods and develops the framework for facade design that is automated and couples simulation engine with optimization algorithms. This framework can be used

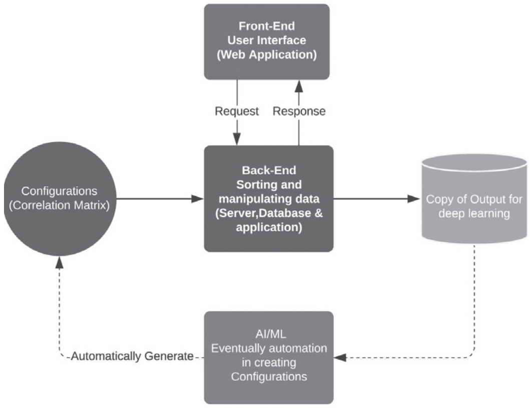

Figure 9: Data Flow Diagram L0 (Author 2019)

as a backend for web or mobile application (or any user interfaces) and, eventually after collecting enough data, the deep learning can be implemented to predict the configurations related to outputs.

Figure 10 represents the whole model development for this facade design tool. The focus of this model and framework is developing the MVP (Minimum Viable Product), which is the core feature to effectively deploy the product to the users. MVP can be part of the process directed toward making and selling this product to the users (Raddof 2014). It works as object in this iterative process of generation, presentation, data collection and analysis and learning. Creating MVP will allow collecting the maximum amount of validated learning data from users.

In this research, MVP is a web application product for users to test and collect data for a next development that needs big data for deep learning. This study mainly focuses on developing a framework as a back end and a simple front end for users' interface. Figure 9 shows the level 0 data flow diagram (DFD), which is known as a context diagram and shows a data system as whole and the way it interacts with external entities.

In order to develop the product based on this research, test the iteration, and collect the results and outputs, we need a user interface as the front-end that is connected to a developed backend. For this purpose, the MVC method is applied. MVC is a software and application design pattern used for developing interfaces and stands for Model, View, Controller as three separated interconnected elements. MVC is a lightweight highly testable framework as compared to traditional ASP.NET web forms. This method separates content from presentation and data processing from content. In other words, design pattern keeps the display and data separate to allow each to change without affecting the other and enable full control over the rendered HTML. So, the main advantages of the MVC

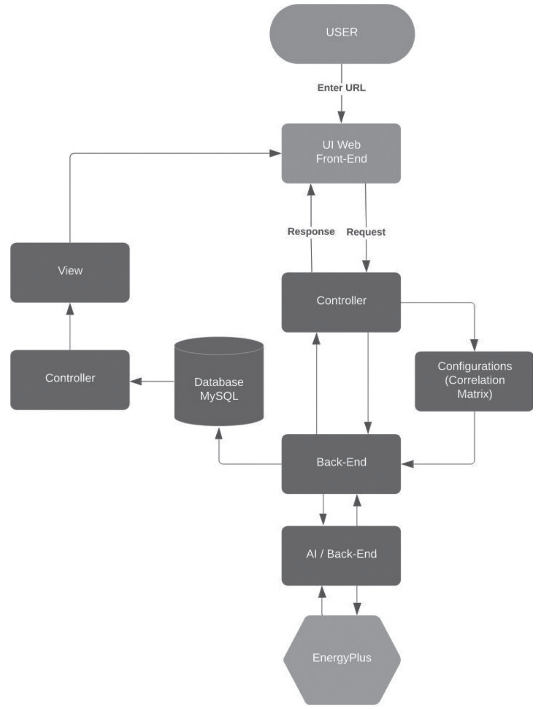

Figure 10: MVC flow chart (Author 2019)

method are providing clean separation of concern (SoC) and enabling test driven development (TDD).

Model represents the shape of data of the application. It is a central component that directly manages the data, logic, and rules of application and is independent of the user interface. It receives user input from the controller. View is a user interface that displays data, so users are able to modify it. View actually works as the front-end in this case. All result representations, such as charts, diagrams, tables, or any other specific forms can be displayed here. Controller handles the user request and renders the appropriate view with the model data as a response. In other words, it accepts inputs and data and converts them to commands for the model or view. Figure 10 illustrates MVC architecture and the flow of users' requests and responses in this case study. The interaction between the front end and back end with simulation engine and creating the configurations.

CONCLUSION AND FUTURE WORK

This research discussed the role of simulations and optimization in the design decision-making process. Then, a novel performance-based facade design framework was described, where different performance criteria and variables have been defined for achieving energy efficiency, occupant comfort, and cost optimality. The framework has been implemented by coupling

EnergyPlus as a simulation engine, and custom scripts using Python programing language. Then a user interface was developed with an MVC method to serve as a front end, in order to test and collect data. The research describes the components and functionality of this framework in detail, as well as the two-step optimization technique. A case study for a test cell was presented, illustrating how the framework is used to test a variety of design possibilities.

Future work will focus on the application for facade design in different climates and developing the user interface. In addition to developing the user interface and web application, this product will be developed to accept any IDF files, the users will be able to choose their variable for optimization and then the rest of the process will be automated and results represented.

After releasing the open source application, other important developments for this research will be collecting the data for implementing deep learning on the data and outputs from different iterations, so as to enable the correlation matrix to generate automatically.

REFERENCES

Aksamija, Ajla. 2013. *Sustainable Facades: Design Methods for High-Performance Building Envelopes.* New York: John Wiley & Sons.

Attia, Shady, Mohamed Hamdy, William O'Brien, and Salvatore Carlucci. 2013. "Assessing Gaps and Needs for Integrating Building Performance Optimization Tools in Net Zero Energy Buildings Design." *Energy and Buildings* 60 (May): 110-124.

Evins, Ralph. 2013. A Review of Computational Optimization Methods Applied to Sustainable Building Design. *Renewable and Sustainable Energy Reviews* 22 (June): 230-245.

Evins, Ralph, Philip Pointer, and Stuart Burgess. 2012. "Multi-Objective Optimization of a Modular Building for Different Climate Types." In *First Building Simulation and Optimization Conference, Loughborough*, 173-180.

Fanger, P.O. 1970. *Thermal Comfort. Analysis and Applications in Environmental Engineering.* Copenhagen: Danish Technical Press.

Gosavi, Ab. 2015. *Simulation-Based Optimization*. Berlin: Springer.

Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors." arXiv preprint arXiv:1207.0580.

Huang, Yu, and Jian-lei Niu. 2016. "Optimal Building Envelope Design Based on Simulated Performance: History, Current Status, and New Potentials." *Energy and Buildings* 117 (April): 387-398.

Ioffe, Sergey, and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." arXiv preprint arXiv:1502.03167.

Li, Xiwang, Jin Wen, and Er-Wei Bai. 2016. "Developing a Whole Building Cooling Energy Forecasting Model for On-Line Operation Optimization Using Proactive System Identification." *Applied Energy* 164 (February): 69-88.

Machairas, Vasileios, Aris Tsangrassoulis, and Kleo Axarli. 2014. "Algorithms for Optimization of Building Design: A Review." *Renewable and Sustainable Energy Reviews* 31(March): 101-112.

Nguyen, Anh-Tuan, Sigrid Reiter, and Philippe Rigo. 2014. "A Review on Simulation-Based Optimization Methods Applied to Building Performance Analysis." *Applied Energy* 113 (January): 1043-1058.

Ostergård, Torben, Rasmus L. Jensen, and Steffen E. Maagaard. 2016. "Building Simulations Supporting Decision Making in Early Design–A Review." *Renewable and Sustainable Energy Reviews* 61 (August): 187-201.

Ries, Eric. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Books.

Shi, Xing, Zhichao Tian, Wenqiang Chen, Binghui Si, and Xing Jin. 2016. "A Review on Building Energy Efficient Design Optimization from the Perspective of Architects." *Renewable and Sustainable Energy Reviews* 65 (November): 872-884.

Wang, Haidong, and Zhiqiang John Zhai. 2016. "Advances in Building Simulation and Computational Techniques: A Review Between 1987 and 2014." *Energy and Buildings* 128 (September): 319-335.